Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives

Oren Zuckerman
MIT Media Laboratory
20 Ames Street E15-020
Cambridge MA 02139 USA
orenz@media.mit.edu

Saeed Arida
MIT School of Architecture
77 Mass Ave. N51-332
Cambridge MA 02139 USA
sarida@mit.edu

Mitchel Resnick
MIT Media Laboratory
20 Ames Street E15-020
Cambridge MA 02139 USA
mres@media.mit.edu

ABSTRACT

This paper introduces a new framework for thinking about tangible interfaces in education, with specific focus on abstract problem domains.

Manipulatives are physical objects specifically designed to foster learning. We offer a new classification of Manipulatives: "Froebel-inspired Manipulatives" (FiMs) and "Montessori-inspired Manipulatives" (MiMs). We argue that FiMs are design materials, fostering modeling of real-world structures, while MiMs foster modeling of more abstract structures. We show that our classification extends to computationally enhanced versions of manipulatives.

We present Digital MiMs – computationally enhanced building blocks. We describe two prototypical members of the Digital MiMs class: FlowBlocks and SystemBlocks, physical, modular interactive systems that serve as general-purpose modeling and simulation tools for dynamic behavior. We present findings from qualitative studies, and conclude that digital MiMs are accessible to young children, engaging, and encourage learning of abstract structures of dynamic behavior through an iterative process of hands-on modeling, simulating, and analogizing.

Authors Keywords

TUI, Digital manipulatives, Simulation, Education, Toys.

ACM Classification

H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

Figure 1: A "normal distribution" simulation

INTRODUCTION

Looking back at the history of Manipulatives, two people stand out as the great thinkers and designers of the field: Friedrich Froebel [2, 6], who designed the "Froebel Gifts" for his Kindergarten, and Maria Montessori [12, 13], who created the "Montessori Materials" for her Casa de Bambini.

Classification of manipulatives

Until now, the physical objects designed by Froebel, Montessori, and others have been collectively called "Manipulatives". We offer a new classification: "Froebel-inspired Manipulatives" (FiMs) and "Montessori-inspired Manipulatives" (MiMs).

FiMs are building toys, enabling children to design real-world things, objects, and physical structures. For example, using wooden blocks to build a structure that resembles a castle. MiMs are a set of building blocks too, but are primarily focused on modeling conceptual, more abstract structures. For example, arranging Cuisenaire rods in different configurations that make numerical proportions

	FiMs	MiMs
TRADITION	Froebel gifts	Abacus Montessori materials
PRODUCTS	Building blocks K'nex LEGO bricks LEGO Mindstorms Tinkertoy Zome	Base Ten Blocks Cuisenaire Rods Fraction Tiles
RESEARCH	ActiveCube* Bitball, Beads* Block Jam* Electronic Blocks* Topobo*	**Digital MiMs** FlowBlocks SystemBlocks

Table 1: FiMs and MiMs Classification

*see Related Work section for details

more salient. FiMs encourage design while MiMs encourage more limited exploration, striving to make abstract concepts more salient.

There are many similarities between FiMs and MiMs. In the hands of a good teacher, FiMs can enable exploration of abstract concepts and MiMs can serve as design material, but clearly, the center of gravity of each class is different. At their core, FiMs focus on design of real-world objects, while MiMs are concerned with modeling of more abstract structures.

The Tangible Advantage in Abstract Domains

Abstract concepts are hard to learn. The advantage of tangible interfaces as a teaching tool for abstract problem domains is threefold:

(1) Sensory engagement – the natural way children learn, engaging multiple senses (in this case touch, vision, auditory) in a constructive process.

(2) Accessibility – dramatically improves accessibility to younger children, to people with learning disabilities, and to novices.

(3) Group learning – provides a multi-hand interface, does not give the control to one person, facilitates natural group interaction, and promotes group discussion.

Physical objects have been traditionally used in kindergartens and elementary schools to introduce young learners to abstract concepts such as quantity, numbers, base ten, fractions etc. Abstract concepts of dynamic behavior, involving change and behavior over time, make the learning challenge even harder.

In our evaluations, some four-year-olds used SystemBlocks to simulate dynamic accumulation, while six and eight year olds used FlowBlocks to collaboratively model probabilistic behavior. A ten-year-old girl said: "I am a person that likes to do things with my hands. With software on the computer, it's always just clicking buttons and inserting numbers and then a window opens and you see the result. With the blocks I can feel what I'm doing, I can see the flow."

Digital Manipulatives

Digital Manipulatives (DM) [19] are computationally enhanced versions of physical objects, created in an effort to expand the range of concepts that children can explore through direct manipulation. DM differ from traditional manipulatives in their ability to model temporal and computational processes. For example, when a child throws the Bitball [19] in the air, the Bitball's embedded computation can measure and plot the ball's velocity and acceleration.

We argue that the FiMs/MiMs classification applies to DM as well (see Table 1 and Related Work section). Furthermore, the classification makes it clear that the majority of current products and research projects are FiMs. We believe there is a great need to strengthen the under-populated field of Digital MiMs.
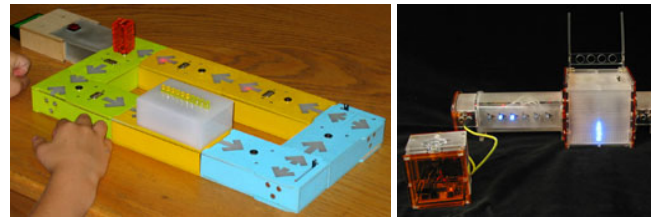

Figure 2: FlowBlocks (left) and System Blocks (right)

Digital MiMs

Digital MiMs are computationally enhanced building blocks systems that support learning of abstract concepts. Digital MiMs involve physical construction, but the result of the construction is a simulation of a generic structure, rather than a real-world example. Digital MiMs can support learning in different domains of knowledge including the mathematics of change, dynamic behavior, probabilistic behavior, and more.

For example, SystemBlocks, one of our two Digital MiMs prototypes, can simulate system dynamics models [4, 22], involving concepts such as rates, accumulation, and feedback. By way of analogy, children see that the same generic structure can represent: water flow through a bathtub, virus spread in a population, bank account savings growth from interest rate, CO_2 pollution growth from emissions, and more. FlowBlocks, our second prototype, can simulate mathematical concepts such as counting and probability, as well as computer-science concepts such as looping, branching, and variables.

Our Digital MiMs maintain coincident and synchronous input/output behavior, meaning that manipulation and simulation occurs at the same space (on the building blocks themselves) and in real-time. Digital MiMs construct abstract structures, but nevertheless are playful tools, facilitating a physical, multi-sensory, interactive experience.

In our evaluations, we observed how 25 children ages 4-11 interact with FlowBlocks and SystemBlocks for an aggregated duration of 40 hours. The children played with the blocks individually as well as in mixed ages groups. Our findings suggest that Digital MiMs are engaging for children, and successfully introduce specific concepts such as rate, accumulation, feedback, and probability to different age groups. Furthermore, we observed how the special characteristics of Digital MiMs encouraged children to make analogies between the simulated abstract behavior and real-life examples that are meaningful to the children. For example, using SystemBlocks, a 10-year-old girl analogized a "water flow through the bathtub" simulation to "books flow through the school library" simulation, and was able to come up with the different variables that influence the flow of books (rate of librarian putting books on bookshelf vs. rate of children taking books away from bookshelf). Using FlowBlocks, a 10-year-old boy created a sequence with a probability junction and analogized the

Figure 3: SystemBlocks simulating bathtub dynamics

simulation to "the chance my mom would say yes when I ask to play video games".

We conclude that digital MiMs are accessible to young children, engaging, and encourage learning of abstract structures of dynamic behavior through an iterative process of hands-on modeling, simulating, and analogizing.

In the next section we present our design process and what we learned from it, including our design guidelines for Digital MiMs. Later we give a hypothetical scenario of children interacting with our prototypes, and report the details of our evaluations, where 25 children interacted with our prototypes for a total duration of 40 hours.

DESIGNING DIGITAL MiMs
In this section we present our design process, list our design guidelines for Digital MiMs, and describe our Digital MiMs prototypes - FlowBlocks and SystemBlocks.

Early Prototype
In 2003 we developed and evaluated an early MiMs prototype. We presented this work at CHI 2003 as a short talk, called "A Physical Interface for System Dynamics Simulation" [31]. This prototype had five building blocks and enabled children to simulate reinforcing and balancing feedback loops by connecting the blocks in different arrangements. Since then we learned about several flaws in the design of that early prototype: the blocks' behaviors were not authentic to a theoretical framework; each building block was not meaningful on its own; and there were no method to help children make analogies. In addition, we wanted to improve the technical infrastructure and the physical design. Nevertheless, the early prototype was instrumental to our design process, and enabled us to evaluate the concept with children, colleagues, and members of the HCI community.

Design Guidelines
The following are a set of "Digital MiMs Design Guidelines" we developed throughout our design process. Guidelines 1-4 define the core of a Digital MiM, and are useful to differentiate MiMs from FiMs. Guidelines A-D are general design criteria, some specific for Tangible Interfaces [26].

(1) Generic structures vs. real-world objects – the building blocks should enable modeling of generic, abstract structures, as opposed to real-world objects (e.g.

model the generic behavior of exponential growth, rather than a specific example like a virus spreading).

(2) Level of Abstraction - maintain a high level of abstraction of the constructed simulations and structures, so concreteness would come from a child's analogies rather than a structure's visual form (e.g. the constructed creation should not look like something familiar from real-life).

(3) Semantic Association - maintain a rigorous and theoretically grounded association between the building blocks and their underlying meaning (e.g each block should represent a specific mathematical operation or concept).

(4) Encourage analogies - provide a method to concretize a general structure using a variety of examples (e.g. a way for children to write a meaning on each block, or a way to place pictures on a simulated structure to highlight the analogy to real-life situations).

(A) Modularity - develop a simple set of building blocks that can be connected in a variety of ways, enabling children to construct different models.

(B) Multi-sensory representations – provide multiple representations for the simulated behavior (such as light, sound, numeric display, graph) to support different styles of learning.

(C) Coincide i/o – manipulation and simulation occur at the same space (e.g. a child constructs and tweaks the simulation using only the blocks, with no need for an additional GUI).

(D) Synchronous i/o – manipulation and simulation occurs at the same time (e.g. a child can tweak a simulation and see the result in real-time).

The SystemBlocks Digital MiM
We created SystemBlocks to give children a hands-on simulation tool to explore systems concepts. SystemBlocks is designed to surface concepts related to rates, accumulation, simultaneous processes, positive feedback, and negative feedback.

The Building Blocks
There are four types of components to SystemBlocks: Stocks, Flows, Variables, and Probes. Stocks are accumulators, representing their value with a vertical line of LEDs. Flows are rates, representing their value with light moving on a horizontal line of LEDs. Flows are connected to a Stock, either as inflow (increase the amount in the Stock) or outflow (decrease the amount in the Stock). Variables are the valves of the Flows, controlling the rate of each Flow. Probes are additional representations, on top of the default LEDs representation. For example, a sound Probe maps the stock level to MIDI notes and the Flows rate to tempo. A line-graph Probe maps the stock level to a line graph height and the Flows rate to the line-graph slope.

Hypothetical Scenario – Playing with SystemBlocks

Jack, a 10-year-old boy has choice time at his 5th grade class. He chooses to play with SystemBlocks, and the teacher arranges the blocks in one of the classroom corners. Jack is already somewhat familiar with SystemBlocks. He saw the teacher demonstrating them a few weeks ago, and last week his friend Daniel chose them for his choice time, but Jack never used them by himself. He starts by setting up one of the examples from the "Systems Around Us" guide. He browses through the different examples until he sees a picture of playing fields in a park. He likes baseball, so he picks this example. He takes the relevant cards out from the picture cards box, and stares at them. There are three picture cards: "people arriving at the playing fields", "number of people currently at the playing fields", and "people leaving the playing fields". Jack knows he needs to place the cards in a certain order on the blocks before simulating. He is confused, not sure what order to place them. He decides to look back at the "bathtub example" he saw the teachers demonstrating: "water flow into bathtub", "level of water in the bathtub", and "water flow out from bathtub". He remembers how he should place them (see figure 3). Now, he thinks he knows how to place the playing fields cards. He places them, and starts to simulate. Turning the dial on the inflow Variable block, he sees the Flow block lights moving towards the Stock, "those are the people arriving at the playing fields" he says to himself. He watches how the light level in the Stock block increases, "this is probably the number of people currently in the playing fields". The light in the Stock block continues to accumulate, and before it reaches the top, Jack turns the dial on the outflow Variable block all the way. Now, he sees the second Flow block lights moving away from the Stock, at a faster speed than the inflow, "those are the people leaving the playing fields". To his surprise, the light level in the Stock block starts to decrease. "Hmmm…more people are leaving than arriving, that's why there are less people in the park…like when it's getting dark and all the younger kids have to go". Jack is satisfied, and looks for another example. He selects the "air pollution" example, because he remembers his mom was talking about buying a new car that pollutes less than the current one they have. This time he decides to connect sound to hear the simulation on top of the built-in moving lights.

More Advanced SystemBlocks Simulations

SystemBlocks is a physical implementation of the Stocks & Flows modeling language [4, 22], and therefore is a general-purpose modeling tool. Additional Stocks, Flows, and Variables can be added without compromising performance. SystemBlocks can simulate first-order, second-order, or "any-order" positive and negative feedback, by connecting Stocks back to Flows [30]. SystemBlocks can simulate supply-chain or food-chain models by connecting Stocks and Flows in a sequence.
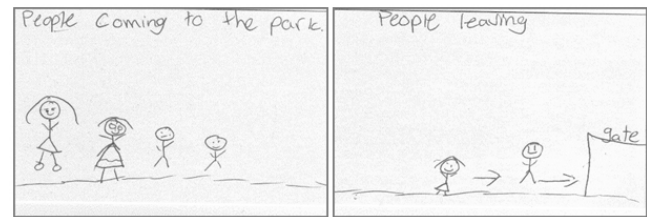

Figure 4: personal analogy of a 10-year-old girl

SystemBlocks can represent simulations using the line-graph Probe, a valuable medium for more complex simulations, such as exponential growth, exponential decay, goal seeking, and a variety of oscillating systems.

Encouraging Analogies with SystemBlocks

Structures created with SystemBlocks do not encourage real-world association. This invites the child to manipulate a generic behavior, rather than a specific real-world example. We designed a specific method to encourage analogies to the simulated generic structures. This method connects a simulation with a real-world example, without compromising the generality of the blocks. Our method includes sets of picture cards, and a designated "card holder" mounted on each of the blocks. Children can easily place cards on top of the blocks and swap different examples (see figures 3, 4, 9). At first, it might seem that this method violates our first two design guidelines (generic structures not real-world objects, and high level of abstraction). In fact, the ephemeral nature of the picture cards method enhances the generality of the blocks, by inviting children to quickly swap a variety of real-world examples, and make their own analogies. In our evaluation interviews, we observed how children quickly swapped 5-10 examples from different domains during less than 5 minutes of simulation.

The FlowBlocks Digital MiM

We created FlowBlocks to enable children to manipulate abstract structures of dynamic processes. We wanted to design a system that could "grow with the child", starting with preschoolers making their first steps in counting and quantity, and continuing with high school or college students struggling with calculus and statistics. FlowBlocks is designed to simulate concepts related to counting, probability, looping and branching.

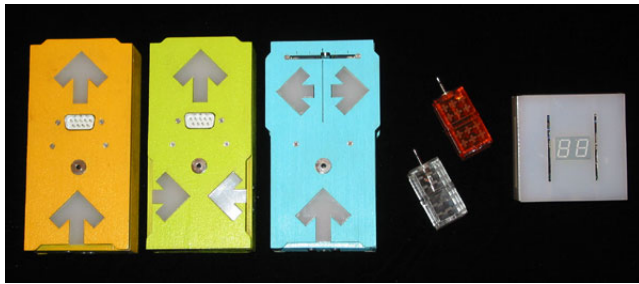**Figure 5: connecting FlowBlocks (left) forming loops (right)**

Figure 6: Left to right – Yellow "Straight" Path, Green "Turn" Path, Blue "Probability Fork" Path, "Faster" and "Slower" Rules, and a "Counter" Probe

The Building Blocks
There are four types of components to FlowBlocks: Paths, Generators, Rules, and Probes. Path blocks connect to each other using magnetic connectors. Light is representing the dynamic process, by "moving" from one Path block to the next in a sequential process. The Generator block supplies power to the other blocks, and has a button that sends out the first light. Rules blocks are small units that plug into the Path blocks and change the internal rule programmed into that block. For example, Rules can make the light go faster or slower. Probes are additional representations, such as the "Counter" Probe that numerically counts how many times the light passed through it.

Hypothetical Scenario - Interacting with FlowBlocks
Three 8-year-old girls, Valerie, Suzanne, and Karen, gather around a table with a set of FlowBlocks. Valerie takes the Generator block, turns it on, and brings it close to a Yellow "Straight" Path block. The magnetic connectors pull the two blocks together, and the Yellow "Straight" Path block flash to signal it was automatically turned on. Valerie clicks the button on the Generator and watches the light travels through the Yellow "Straight" Path block she just connected. Meanwhile, Suzanne played with Green "Turn" Path blocks and created an "L" shape by snapping two of them together. "Hey, Valerie, maybe we can connect my green blocks to your yellow". Together, they bring the blocks closer until they snap. Valerie clicks the button again, and they watch the light travel straight through the Yellow, take a turn through the Green, and another turn through the second Green. Karen snaps another Yellow and in no time they connect all the 25 Path blocks in a long and winding sequence. This time, Karen clicks the button to send the light. When it reaches a Blue "Probability Fork" Block, the light turns to the left instead of to the right, where the next block is connected. They stare at it for a while. Suzanne starts to tinker with the slider mounted on the Blue "Probability Fork" Block's surface, noticing she can move it from side to side. She moves it all the way to the right and clicks the button again. They watch the light with anticipation and scream with joy when the light turns to the right direction. Now the three of them quickly move all the Blue blocks' sliders along the path, and the light travels successfully through all the blocks along the sequence. They are satisfied and immediately look for the next thing to do. Karen looks for more blocks to add

to their sequence, and finds one of the small Rule units. "It says faster on it" she says to her friends, and snaps it into the top surface of one of the Path blocks in their sequence. Valerie clicks the button and they watch with anticipation, waiting for the light to reach the Faster Rule. When the light hits the Rule, it speeds up to a new constant speed and travels at that speed throughout the sequence. They plug all the Faster and Slower Rules Blocks along the sequence, watching the light go faster or slower at each sub-sequence. While the light is traveling, Suzanne is tinkering again with the Blue block's slider, making the light disappear by turning left to nowhere. "Hey, what if we connect more blocks to the left side as well?" They quickly snap-out blocks from their long sequence to construct a new one from both sides of the Blue "Probability Fork" Block. "Hey, what if we make a square and connect this Yellow one back to this Green one?" says Valerie. They quickly snap blocks in and out, trying to form a square. When they are done, Valerie clicks the button and they all watch the light. "It goes round and round" Valerie says. "Do you think it will ever stop?" says Karen. "Let's make it go faster" says Suzanne, and plugs a Faster Rule. "Look, it's getting faster and faster" says Valerie, "it makes my head spin…let's make it go slower again".
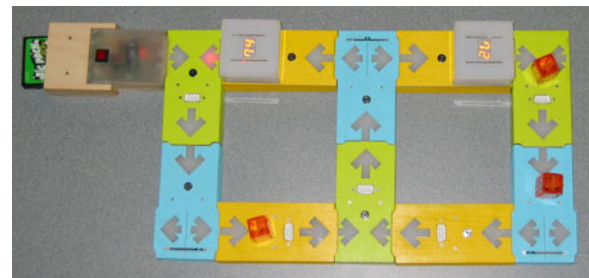

Figure 7: Two loops with 50% probability measurement

More Advanced FlowBlocks simulations
FlowBlocks can simulate many more models. Figure 7 shows a "two loops" arrangement with a Blue probability block serving as a "junction". In this simulation, the probability slider is set at the middle for 50% probability to turn right or left. Two display probes are plugged on each side of the Blue block, each counting how many times the light has passed through. In figure 7 the display probes show "74" and "72", as the simulation has already been running for some time.

Figure 1 shows a tree structure simulating "normal distribution". Each Blue "Probability Fork" Block in the tree structure is set at 50% probability. Children can experience it by keeping the Generator button pressed, which in turn sends a continuous stream of numbers into the blocks. The light will travel in the tree structure, and the display Probes will count how many lights reached each branch of the tree. The result of this simulation will be the "bell shape" or "normal distribution" curve, noticed even with just 3 branches.
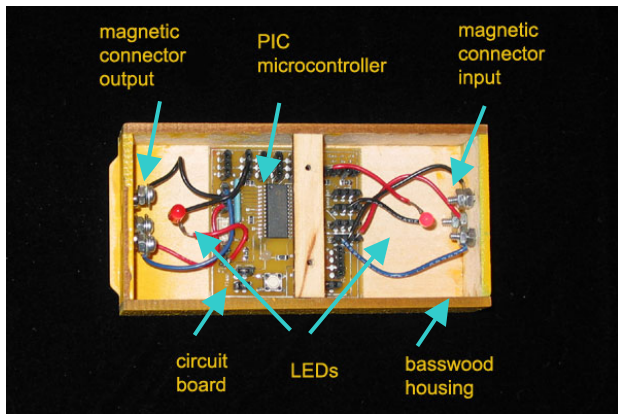
Figure 8: Inside a Yellow Path block

TECHNICAL IMPLEMENTATION

FlowBlocks and SystemBlocks are made from laser-cut Plexiglas or basswood parts, which are later glued together. The basswood blocks are sanded and airbrushed.

The hardware is a 20 MHz PIC16F876 microcontroller mounted on a dedicated printed-circuit-board that handles power distribution, local processing, A/D conversion, and 16bit serial network communication (see figure 8). A single 6V battery powers the whole system. The firmware is written in Logo [15] and runs on a Logochip PIC16F virtual machine [23]. A 16bit number system was implemented to enable non-integer calculations (limited to 2 digits after the decimal point), to ensure accuracy when simulating exponential decay and similar simulations.

A mix of custom-made and commercially available connectors transfers power, ground, and data between the different blocks. The custom-made connectors snap the FlowBlocks' Path blocks together using neodymium pressure-formed magnets (see figure 8). The other blocks use a variety of commercially available connectors, including 2.5mm mono, 3.5mm stereo, and DB9.

5mm LEDs are used throughout the system to provide the "moving lights" representation. 7-segment LED display units are used to provide the number display. The Cricket bus-device MIDI hardware is used to provide the sound representation. Hyundai graphic display module is used to provide the line graph representation.

System Limitations

When enthusiastic children used our systems, several problems occurred consistently and pointed us to our system's limitations. The Acrylic used in SystemBlocks tend to break, the wood we later used in FlowBlocks was much more robust. The single battery source in FlowBlocks sometimes failed to supply power when a new block was added to a sequence (some sequences reached 15 blocks). FlowBlocks custom-made magnetic connectors experienced failures in signal transfer. We plan to address these problems in future prototypes.

EVALUATION

We conducted separate evaluations for FlowBlocks and SystemBlocks. In total we observed 25 children interacting with Digital MiMs for a total duration of 40 hours.

Our research approach was a qualitative one. We used a clinical-interview approach where the interviewer presents brief, standard tasks to the students, and then probes the students' understanding based upon their response to the tasks. The standard set of probes were loosely structured and designed to follow up on what the students said.

SystemBlocks study

The SystemBlocks studies focused on individual sessions with two age groups, 5th graders (10-year-olds) and preschoolers (4-year-olds).

5th grade study

The study with 10-year-olds was performed at two public schools in Massachusetts with ten children, four girls and six boys. We conducted 2-3 individual sessions with each child, when average session duration was 45 minutes. All sessions were videotaped for later analysis. The tasks in each of the individual SystemBlocks sessions were presented by the interviewer but performed solely by the student: (1) mapping of picture cards onto a simple inflow-stock-outflow structure, (2) simulating the model and analyzing net-flow dynamics using moving lights and sound, (3) analyzing net-flow dynamics using real-time line graph, (4) analyzing models with simple positive-feedback loop, (5) generating analogies of simple systems.

The 10-year-olds easily associated the moving lights to flow or accumulation processes. They quickly mapped the example picture cards onto the simulated model, and were able to generate their own analogies. In the process of mapping picture cards and simulating a new example, several misconceptions and tendencies surfaced. For example, Julian, a 10-year-old boy, drew a basketball example (see figure 9). Julian probably analogized from examples he simulated earlier, such as the bathtub example ("water flow into bathtub", "level of water in the bathtub", "water flow out from bathtub"). In the bathtub example, there are simultaneous processes. Water can flow into and out from the bathtub at the same time. The higher the rate of water flow in or out, the higher the rate of change in the bathtub water level. In Julian's example, the processes are sequential, rather than simultaneous. First he gets a basketball, then he practice, then he gets better. "Get a basketball" does not necessarily cause "practice" to increase, and "how good you are" does not necessarily cause "practice" to decrease. Julian simulated his example, and after a few minutes of interactive simulation he asked if he could change his example. Interacting with SystemBlocks helped him see, by himself, that the casual model in his example does not apply to the generic structure he simulated. With the interviewer's encouragement Julian tried to figure out how to change his

Figure 9: Julian's basketball example - (a) 'get a basketball', (b) 'practice', (c) 'how good you are'

example. This time he came up with: "time you spend playing basketball", and "how good you are". For the outflow he wasn't sure, and suggested "time you spend doing other things". This time, his example works. The more time he spends playing, the better he becomes. Six out of the ten 5th graders repeated Julian's tendency for sequential causality rather than simultaneous causality. This tendency might inform us that children are growing up aware to the sequential processes around them (found also in any narrative structure), but with less understanding of the simultaneous processes and more complex systems in their lives.

We observed additional tendencies and misconceptions [29]. For example: preferring quantity (stock) over process (flow) when mapping an example (like placing the "water in bathtub" card before the "water flow into bathtub" card), preferring inflow over outflow when adjusting a simulation (like trying to maintain financial balance by increasing earnings vs. decreasing spending).

Preschool study
The exploratory study with 4-year-olds was performed at the MIT child care center with five children, three girls and two boys. We conducted one individual session with each child, with an average session duration of 20 minutes. All sessions were videotaped for later analysis. The tasks in these sessions were: (1) filling a beaker with real water from the faucet at different rates, (2) mapping the faucet-and-beaker example using picture cards onto an inflow-stock structure, (3) analogizing the moving lights to water. (4) Mapping other picture cards examples onto the inflow-stock-outflow model (examples were cookies and money).

Out of the five children, three could analogize the moving lights to water, and two could not. The ones that could not said "the lights are blinking". They could not recognize the moving illusion of the blinking LEDs, and lost interest quickly. The three children that could analogize moved on to the cookies example. We placed the following picture cards on the SystemBlocks: "baking cookies", "number of cookies made", and "eating cookies". The 4-year-olds successfully simulated the model while describing with joy how they bake the cookies and other kids eat the cookies. Charlie, a 4-year-old boy, pointed at the lights in the stock as the level increased, and counted out loud "one cookie, two cookies, three cookies, four cookies". When he turned the outflow dial to increase the "eating cookies" rate he waited until the light level in the stock decreased completely and said "no more cookies".

SystemBlocks study summary
Throughout both sessions (10-year-olds and 4-year-olds), the children that could analogize the moving lights were very engaged and hands-on with SystemBlocks. The tangibility aspect of System Blocks promoted discussion, and was effective in surfacing students' mental models and exposing misconceptions. Lisa, a 10-year-old girl, compared the experience she had with SystemBlocks to working with a computer at school: "I am a person that likes to do things with my hands. With software on the computer, it's always just clicking buttons. With the blocks I can feel what I'm doing. I like it much more." Jim, a 10-year-old boy, who had previous experience with a software simulation tool called STELLA [24], said: "I like the blocks much more than STELLA. With STELLA, you click buttons and insert numbers and then a window opens and you see the result. With the blocks, I can see the flow, I can change this dial and see the lights move faster." When asked about the multiple medium used in SystemBlocks, Lisa said: "I think the lights and the sound are very helpful. The graph is also helpful, but I like the sound better. Starting with the lights, then hearing the sound, and then seeing the graph was great."

FlowBlocks Study
The FlowBlocks evaluation focused on individual as well as small-group sessions with mixed-age groups (three children per group), ranging in age from 6-11 year-olds. The study was performed at MIT's Westgate community center with ten children, five girls and five boys. The average session duration was 45 minutes. All sessions were videotaped for later analysis. These sessions were designed to be more open-ended play. During the sessions, we looked for the following activities: (1) building sequences with various speed changes, (2) forming a loop with increasing speed, (3) using the Blue probability block to direct sequences, (4) using the Probes to perform counting and probability measurements, (6) reacting to the "normal distribution" statistics simulation, constructed by the interviewer.

All the children were able to quickly form sequences by themselves, figuring out the way the Path blocks connect to each other. The magnetic connectors were very effective in helping the children "guess their way" though the construction process. The sequences they created were directly influenced by the set of blocks they were offered. When all the blocks were available to them, they tried to use all of them by forming a long and winding sequence. With fewer blocks square-shaped loops were starting to form. There was a clear difference in the amount of time it took the younger children (6-8 year-olds) to form a loop vs. the older children (9-11 year-olds). But after the first loop was formed, all ages could quickly form new ones. Most of the children called the loops "squares" but some called them "circles", even though their visual shape is a square. The children were fascinated with the light traveling through the loops. Tina, an 8-year-old girl said "it's going round and round, do you think it will ever stop?" Gaia, a 7-year-old girl added "there is no

end". All the children loved to plug the Rules blocks in and out of the Paths, and see the immediate change in the simulation. Michael, a 9-year-old, plugged both Faster and Slower Rules into his loop. When asked to make the light go faster, he quickly took one of the Slower Rules out, instead of adding another Faster one. This demonstrates ability to think in "mediating causality" terms [17]. The same action was observed with all the other children, with no difference in their age. When Christopher, a 10-year-old boy saw the display Probes, he quickly plugged them into a Path block and formed a loop around it. He simulated it, and said: "awesome, it's counting". By mistake he clicked the generator button again, which sent another light to the loop. "Look, now it's counting by two" he said. In a different session, Sean, a 6-year-old boy, was bored when his older brother was forming a sequence. He took a few blocks and started to form a sequence by himself. When asked if he wanted the Generator block to simulate his structure, he said: "no, I don't need the light, I use my mind to think it up". Can FlowBlocks help children run simple simulations in their minds?

When Karen, an 8-year-old girl formed a sequence with the Blue probability block, she insisted there is a pattern in the number of times the light is turning to each side. Her guess for the pattern was "first two to the right and then two to the left". After some observation she was confused. Her friend Suzanne suggested to measure it using the Probes. They plugged two display Probes in and watched the numbers increase in both Probes. They could still not figure out the pattern, but saw that over time, the numbers increase in almost similar amounts. They watched the numbers for almost two minutes until both Probes reached their maximum (2-digit displays, limited to 99). The Probes reached their max in a difference of 2 seconds. "The difference was really small, they reached almost together". They decided to try the whole thing again, to see what will happen the next time. In a different session, Alexander, 11-year-old boy, said the Blue block is his favorite block "because it is unpredictable". When asked about other unpredictable things in his life, he said: "It is like the chance for my mom to say yes when I ask her to let me play video games". He took the Blue block and moved the slider to 25% right turn vs. 75% left turn. "This is my chance to get a yes", he said, "there is more chance she will say no than yes".

During the sessions, the younger children tried to form real-world structures, like letters, numbers, triangles. Gaia asked if she could build her name and run the light through it. Karen held a loop sequence up and peeked through its middle hole, pretending to hold a camera. It seems the younger children felt a need to concretize the structures in a visual way. We think it is better to design Digital MiMs in such a way that will not encourage any specific, real-world visual forms, only to increase the chance that children will create analogies to the abstract processes, rather than the physical form.

RELATED WORK

Several software-based simulation environments for dynamic behavior, such as STELLA [22, 24], Vensim [27], Model-It [7], and StarLogo [20], have encouraged us to design a tangible interface for dynamic behavior. Ishii's Tangible Bits vision [26] helped us clarify our user interface approach.

Many research labs have developed and researched Tangible User Interfaces (TUIs) in education. Several UI approaches have been taken. In one approach, the TUI serves as an input device to a standard computer, and the output is displayed on a separated GUI. Projects in this category include: AlgoBlocks by Suzuki and Kato [25] – a block-based interface to create LOGO programs; ActiveCubes and TSU.MI.KI from Osaka University [8, 9] – a cube based interface for the construction of and interaction with 3D environments; Block Jam by Sony's Newton-Dunn et al. [14] – a block based interface for music composition. In another approach the output is projected from a standard computer on top of the TUI input device. Projects in this category include: The TICLE project by Scarlatos [10] – augmenting Tangram puzzle construction with computer-vision-based "guide on the side"; and CircuiTUI by Chak and Ishii - an electronics circuit-design tool based on the SenseTable technology [16]. Another approach is TUI as a front-end to a ubiquitous computing system, creating an interactive space or immersive environment. Projects in this category include MIT's Kids Room [kidsroom], University of Maryland's Physical Programming and Story Rooms projects [programming and story rooms], Philips Design's Pogo [Philips Design], and Sweden's University of Umea's Ely the Explorer [ely].

Our prototypes belong to a different approach, in which computation is embedded inside the physical object, creating a TUI that serves both as input and output, with no standard computers involved. Research projects in this category can be best described as extending the work of Froebel and Montessori, presenting new computational based learning materials. Projects in this category include: the first digital manipulatives, the Bitball and Beads by Resnick et al. [19] – a ball that displays its acceleration using colored light, and light-emitting beads that form different light patterns when rearranged; Tangible Programming Bricks by Tim McNerney [11] – computational LEGO bricks to explore sensing and computation; Curlybot [5] and Topobo [18] by Ishii's group – "programming by example" systems to explore physical motion in 2D and 3D; and the electronics Duplo Blocks by Wyeth and Purchase [28] – computational Duplo blocks to explore logic and computation through play.

Table 1 shows the FiMs/MiMs classification for traditional manipulatives, popular toys/learning materials, and the research projects that focus on Manipulative-style learning. Following is a detailed explanation of our classification in the research category. We used our Digital MiMs Design Guidelines (DMDG) to differentiate between the classes. See

the number of the DMDG guideline we used at the end of each paragraph.

TICLE [10] – a computer-vision system that tracks children's play with the Chinese geometry puzzle Tangram. The system scaffolds the play process with hints in real-time. TICLE focuses on scaffolding the play process with the Tangram puzzle, and in the same way, could scaffold children's play with other manipulatives, either FiMs or MiMs. Therefore, we do not classify it as either FiM or MiM.

Bitball [19] - is a rubbery ball that changes color based on changes in acceleration. Bitball models a specific real-world example of the derivative concept (acceleration/velocity). See DMDG # 1, 2, 4.

Beads [19] - are programmable electronic beads that engage children in creating "one dimensional" dynamic light patterns. The Beads inspired us when we designed FlowBlocks, and in our work we extended some of the Beads underlying themes. Beads focus on the creation of necklaces, and the bead's internal operations do not represent a specific mathematical operation, but was rather designed to make it easier to create engaging light patterns. See DMDG # 1, 3, 4.

Block Jam [14] - is a block interface for interactive music creation. Block Jam developers define it as a Modular Tangible Interface that is "Functionally Homogeneous" vs. "Functionally Heterogeneous" - meaning there is one type of physical artifact with a single function rather than different physical artifacts each holding a different function. Block Jam was not designed to help people understand what the building blocks of a musical sequence are, but was rather designed to make it easier to construct a musical sequence in an expressive process. See DMDG # 1, 3, 4.

Topobo [18] – is a 3D constructive assembly system with kinetic memory enabling people to assemble and animate dynamic biomorphic forms like animals and skeletons. Topobo encourages design and construction of real-world creations such as a walking bug, dog, or moose. See DMDG # 1, 2, 4.

Electronic Duplo Blocks [28] – a LEGO Duplo based programming environment for children that includes sensor blocks, action blocks and logic blocks. Output is presented as a car's physical motion, a sound melody, and a light. The Electronic Blocks model generic behavior rather than specific ones, and maintain a rigorous association between the blocks and their underlying meaning (DMDG # 1, 3). At the same time, they limit the range of abstractions and analogies by using a car and a specific melody. See DMDG # 2, 4.

ActiveCube [8] – a cube-based interface allowing users to build 3D structures in the real world while computer software automatically generates a corresponding 3D virtual model that is displayed on the computer screen. In addition, the computer retrieves similar shapes from its 3D models database, such as an airplane, house, or car. ActiveCube encourages design and construction of real-world objects. See DMDG # 1, 2, 4.

We would like to emphasize that classifying projects to FiMs or MiMs implies nothing on a project's educational value. FiMs and MiMs, just like Froebel's Gifts and Montessori's Materials, can both be excellent learning tools that encourage learning in different ways.

FUTURE WORK

Future design efforts will focus on adding new mathematical concepts to FlowBlocks, such as addition, multiplication and other arithmetic operations. We plan to extend the probabilistic behavior. We would like to expand the range of structures by adding new Path blocks with different angles and curves, as well as expansion to 3D.

We hope to extend the Digital MiMs family and develop new interactive systems that focus on a range of abstract concepts. We plan to continue our evaluations of Digital MiMs both with children and adults, in an effort to understand what concepts are more appropriate for MiMs rather than FiMs, and vice versa.

CONCLUSION

We presented Digital MiMs – a class of computationally enhanced manipulatives that encourage modeling of abstract structures. We described two prototypical members of the Digital MiMs class: FlowBlocks and SystemBlocks, physical, modular interactive systems that serve as general-purpose modeling and simulation tools for dynamic behavior.

Our research suggests that Digital MiMs are engaging learning environments despite their high level of abstraction, and gives children an opportunity to interact with dynamic behavior at the symbolic level rather than the example level. We showed that Digital MiMs, as other TUIs for learning, promote group interaction and discussion. Furthermore, our SystemBlocks 5th grade study suggests that an iterative process of hands-on modeling and simulation provides children with an opportunity to confront their misconceptions about dynamic behavior.

We hope that our research will promote future study in the under-populated field of Digital MiMs

REFERENCES

1. Decortis, F., Marti, P., Moderini, C., Rizzo, A., and Rutgers, J. Disappearing computer, emerging creativity: an educational environment for cooperative story building. Proceedings of IDC 2002.

2. Africano D. et al. (2004). Designing tangible interfaces for children's collaboration. Proceeding of CHI 04.

3. Alborzi, H. et al. (2000). Designing StoryRooms: Interactive storytelling spaces for children. In Proceedings of Designing Interactive Systems, ACM Press, 95–104.

4. Bobick, A. et al. (1999). The Kids Room: A perceptually-based interactive and immersive story environment. In PRESENCE: Teleoperators and Virtual Environments (August 1999), 367– 391.

5. Brosterman, N. Inventing Kindergarten. 1997.

6. Forrester, J W. (1971) "Principles of Systems", Pegasus Communications Inc. MA

7. Frei, Phil. (2000). curlybot: Designing a new class of computational toys. Master's thesis, Massachu-setts Institute of Technology.

8. Fröbel, F. (1826). On the Education of Man (Die Menschenerziehung), Keilhau/Leipzig: Wienbrach.

9. GoKnow Inc. www.goknow.com/Products/Model-It/

10. Ichida H., Itoh Y., Kitamura Y., Kishino F. (2004). ActiveCube and its 3D Applications, IEEE VR 2004, Chicago, IL, USA.

11. Itoh Y. et al. (2004). TSU.MI.KI: Stimulating Children's Creativity and Imagination with Interactive Blocks. In proceeding to the Second International Conference on Creating, Connecting and Collaborating through Computing (C5'04) January 29 - 30, 2004.

12. L.L. Scarlatos (2002). An Application of Tangible Interfaces in Collaborative Learning Environments. In Proceedings of SIGGRAPH 2002, 125-126.

13. McNerney T. (2004). From turtles to Tangible Programming Bricks. In Personal and Ubiquitous Computing, Volume 8, Issue 5 (September 2004), Pages: 326 – 337.

14. Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W., and D'Armour, A. (2001) Physical Programming: Designing Tools for Children to Create Physical Interactive Environments. Proceeding of CHI 2002, CHI Letters, 4(1), 299-306.

15. Montessori, M. (1916). The Montessori Method. English translation (1964). The

16. Montessori, M. (1949). Childhood Education. Henry regency Company, Illinois.

17. Newton-Dunn H., Nakano H., Gibson J. (2003). Block Jam: A Tangible Interface for Interactive Music. Proceeding of NIME-03, Montreal, Canada

18. Papert, S. Mindstorms: Children, computers and powerful ideas. Basic Books, New York, 1980.

19. Patten J., Ishii H., Hines J., Pangarp G. (2001). Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces'. Proceedings of SIGCHI 2001.

20. Perkins, D.N. & Grotzer, T.A. (2000, April). Models and moves: Focusing on dimensions of causal complexity to achieve deeper scientific understanding. Paper presented at the annual conference of the American Educational Research Association (AERA), New Orleans, LA.

21. Raffle, H. Parkes, A. Ishii, H. (2004). Topobo: A Constructive Assembly System with Kinetic Memory. Proceeding of CHI 04.

22. Resnick M. et al. (1998) Digital manipulatives: New toys to think with. Proceeding of CHI 1998.

23. Resnick, M. (1994). Turtles, Termites, and Traffic Jams. Cambridge, MA: MIT Press.

24. Resnick, M., Martin, F., Sargent, R., and Silverman, B. (1996). Programmable Bricks: Toys to Think With. IBM Systems Journal 35, 3, 443-452.

25. Richmond, B. (1992). Introduction to System thinking, High Performance Inc. Lebanon, NH.

26. Silverman B., Mikhak B., Berg R. http://www.wellesley.edu/Physics/Rberg/logochip/

27. STELLA, by HPS Inc. http://www.hps-inc.com

28. Suzuki, H. and H. Kato. (1995). Interaction-Level Support for Collaborative Learning: AlgoBlock -- An Open Programming Language. In Proceedings of CSCL.

29. Ullmer, B and Ishii, H. (2001) Human-Computer Interaction in the New Millenium, John M. Carroll, ed.; © Addison-Wesley, August 2001, pp. 579-601.

30. Vensim, by Ventana Systems. http://www.vensim.com

31. Wyeth Peta, Purchase C. Helen. (2002). Tangible programming elements for young children. Proceeding of CHI 02.

32. Zuckerman O. (2004). System Blocks: Learning about Systems Concepts through Hands-on Modeling and Simulation. MIT Masters Thesis.

33. Zuckerman O., Resnick M. (2003). System Blocks: A Physical Interface for System Dynamics Learning. Proceedings of the 21st International System Dynamics Conference.

34. Zuckerman O., Resnick M. (2003). System Blocks: A Physical Interface for System Dynamics Simulation. In Proceedings of CHI '03, ACM Press, pp. 810-811.